*Załącznik nr 1 do Uchwały nr \_\_\_\_\_\_\_ - 2018/2019 z dnia 25 marca 2019 r.*

*w sprawie wytycznych dla tworzenia i zmian programów studiów pierwszego stopnia, drugiego stopnia*

*oraz jednolitych studiów magisterskich rozpoczynających się od roku akademickiego 2019/2020.*

Opis **zajęć (sylabus)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nazwa zajęć: | Systemy rozproszone | **ECTS** | **5** | | | |
| Nazwa zajęć w j. angielskim: | Distributed systems | | | | | |
| Zajęcia dla kierunku studiów: | **Informatyka** | | | | | |
|  |  | | | | | |
| Język wykładowy: | polski | Poziom studiów: | studia I stopnia | | | |
| Forma studiów: |  stacjonarne   niestacjonarne | Status zajęć: |  podstawowe   kierunkowe |  obowiązkowe   do wyboru | Numer semestru: ……4….. |  semestr zimowy  semestr letni |
|  |  | Rok akademicki, od którego obowiązuje opis (rocznik): | 2019/2020 | Numer katalogowy: | **ZIM-IN-1Z-06L-44\_10** | |
|  | | | | | | |
| Koordynator zajęć: |  | | | | | |
| Prowadzący zajęcia: |  | | | | | |
| Jednostka realizująca: |  | | | | | |
| Jednostka zlecająca: |  | | | | | |
| Założenia, cele i opis zajęć: | Celem przedmiotu jest zapoznanie studentów z podstawowymi zagadnieniami programowania w systemach równoległych i rozproszonych oraz projektowania systemów rozproszonych z wykorzystaniem nowoczesnych narzędzi i technik. Nacisk położony zostanie na nabycie przez studentów umiejętności programowania usług rozproszonych w środowisku heterogenicznym.Tematyka wykładów:  * Wprowadzenie. Podstawowe pojęcia: przetwarzanie współbieżne, równoległe, rozproszone. Definicja systemu rozproszonego. Architektury systemów rozproszonych, systemy klient-serwer. Przykłady: sterowanie rozproszone, Seti@Home, bankomaty, serwer bazodanowy, aplikacyjny, itd., współdzielenie drukarek w sieci. * Cechy systemu rozproszonego: przezroczystość (położenia, dostępu, wędrówki, zwielokrotnienia, współbieżności, trwałości, względem awarii), otwartość (protokoły i interfejsy), skalowalność, problemy synchronizacji (brak globalnego zegara), komunikacja synchroniczna i asynchroniczna. DNS jako przykład systemu rozproszonego. * Sprzęt w systemie rozproszonym. Urządzenia sieciowe (routery, switche, karty sieciowe). Stos protokołów TCP/IP. Pamięć wspólna, a pamięć rozproszona, systemy wieloprocesorowe i wielordzeniowe. Klastry, gridy, chmury, superkomputery. * Równoległość i współbieżność w ramach jednego węzła. Procesy, wątki, mikro-wątki. Mechanizmy współpracy (komunikacji i synchronizacji) między nimi. Zamek, sekcja krytyczna, rodzaje zamków. Model wielowątkowości w języku Python; GIL (*Global Interpreter Lock*). * Komunikacja i synchronizacja między procesami: mechanizy współdzielenia segmentów pamięci, semafory, zmienne warunkowe (monitory). Moduł multiprocessing w Pythonie. * Podstawowe mechanizmy komunikacji i synchronizacji w systemach z pamięcią wspólną a rozproszoną. Komunikacja między węzłami (przekazywanie komunikatów), komunikacja zawodna i niezawodna (TCP/UDP). Problemy wymiany komunikatów (wewnętrzna reprezentacja danych, komunikacja synchroniczna lub asynchroniczna, możliwośc zakleszczenia, itd.). * Komunikacja przy pomocy gniazdek sieciowych (socketów). Ogólne zasady oraz implementacja w języku Python. Oczekiwanie na transmisje z wielu gniazdek. * Wywołanie procedury w systemie komputerowym. Przekazywanie parametrów. Przeniesienie wywołania do systemu rozproszonego. Mechanizmy wywoływania zdalnych procedur (RPC). Przykładowe implementacje: XML-RPC, JSON-RPC, itd. * Architektura web serwisów i mikroserwisów. Architektura REST. Interoperacyjność usług i komponentóœ sieciowych. REST API i Web API w nowoczesnyc hsystemach rozproszonych. * Szczegółowa rola procesów i wątków w systemie rozproszonym. Przykłady klientów i serwerów wielowątkowych. Inne mechanizmy języka Python: biblioteki Asyncore, AsyncIO, Gevent, Tornado. Greenlety i tasklety. Stackless Python. Przykłady zastosowań (gra sieciowa Eve Online, system pocztowy IronPort. * Narzędzia Pythona do tworzenia klastrów (IPython, NSQ, Celery, dispy). * Rozproszony system plików. Omówienie cech i wymagań. NFS jako przykład funkcjonalnego systemu plików w systemie rozproszonym. * Serwer nazw w systemie rozproszonym.   Tematyka ćwiczeń laboratoryjnych:   * Wprowadzenie do języka Python 3, jako podstawy programowania aplikacji rozproszonych. * Wielowątkowe programy w Pythonie. * Tworzenie aplikacji wieloprocesowych. * Komunikacja sieciowa przy pomocy gniazd UDP. * Komunikacja sieciowa przy pomocy gniazd TCP. * Komunikacja sieciowa przy pomocy mechanizmu wywoływania zdalnych procedur: XML-RPC. Szeregowanie (serializacja) danych. * Zastosowania przetwarzania wielowątkowego w serwerach i klientach. * Mechanizmy synchronizacji rozproszonej (semafory, zmienne warunkowe). * Nieblokujące wejście i wyjście. Komunikacja za pośrednictwem WWW.   Tematyka projektów:   * Stworzenie aplikacji sieciowej z wykorzystaniem warstwy pośredniej oprogramowania, demonstrującej ustaloną metodę komunikacji między węzłami, wielowątkowość, mechanizmy synchronizacji i spójności. * Przykładowe aplikacje: gra sieciowa dla dwóch lub większej liczby osób, komunikator internetowy, system składowania i udostępniania plików, system bazodanowy dostępny przy pomocy przeglądarki | | | | | |
| Formy dydaktyczne, liczba godzin: | 1. wykład; liczba godzin ...30...; 2. ćwiczenia laboratoryjne; liczba godzin ...30...; | | | | | |
| Metody dydaktyczne: | wykład, dyskusja problemu, studium przypadków, konsultacje | | | | | |
| Wymagania formalne  i założenia wstępne: | Znajomość sprzętu i oprogramowania wykorzystywania w sieciach komputerowych, znajomość zasad wykonywania programu komputerowego, znajomość zasad działania systemu komputerowego, umiejętność programowania w języku wysokiego poziomu | | | | | |
| Efekty uczenia się: | Wiedza:  01 – zna budowę systemu rozproszonego, potrafi podać jego elementy oraz przykłady oraz klasyfikację  02 – zna i rozumie specyficzne cechy systemu rozproszonego  03 – potrafi opisać operacje wykonywane w systemie rozproszonym, rozumie ich znaczenie  04 – umie programować w języku wykonywanym na pośredniej warstwie oprogramowania  05 – zna i rozumie zasady komunikacji klient-serwer,  06 – zna rodzaje oraz cechy serwerów wykorzystywanych w systemie rozproszonym | Umiejętności:  01 – umie zaimplementować różnorodne technologie komunikacyjne w systemie rozproszonym  02 - umie dokonać optymalizacji komunikacji między komponentami systemu rozprosoznego,  03 – potrafi opracować system rozproszony z wykorzystaniem przeglądarki internetowej, po stronie klienta    04 – rozumie współczesne trendy rozwojowe systemów rozproszonych | Kompetencje:  01 - potrafi zdiagnozować wady systemu rozproszonego | | | |
| Sposób weryfikacji efektów uczenia się: | Egzamin pisemy lub ustny.  Opracowanie projektu podczas zajęć laboratoryjnych. | | | | | |
| Forma dokumentacji osiągniętych efektów uczenia się: | Sprawozdania projektowe.  Egzaminy pisemne lub protokół z przeprowadzenia egzaminu ustnego z ocenami. | | | | | |
| Elementy i wagi mające wpływ  na ocenę końcową: | **Ćwiczenia laboratoryjne – 10%, zadania projektowe – 40%, egzamin – 50%** | | | | | |
| Miejsce realizacji zajęć: | Wykład – dowolna sala, ćwiczenia laboratoryjne – laboratorium komputerowe. Preforwana jedność miejsca. | | | | | |
| Literatura podstawowa i uzupełniająca:  Literatura podstawowa:   * M. Gorelick, I. Ozsvald, „Python. Programuj szybko i wydajnie”, Helion, 2015. * G. Coulouris, J. Dollimore, T. Kindberg, „Systemy rozproszone. Podstawy i projektowanie”, WNT, Warszawa, 1996. * A.S. Tanenbaum, M. van Steen, “Systemy rozproszone: zasady i paradygmaty”, WNT, Warszwa, 2006. * A.S. Tanenbaum, „Rozproszone systemy operacyjne”, Wydawnictwo Naukowe PWN, Warszawa, 1997. * W.R. Stevens, „Programowanie zastosowań sieciowych w systemie Unix”, WNT, Warszawa, 1998.   Literatura uzupełniająca:   * T. Ziade, „Rozwijanie mikrousług w Pythonie”, Helion, 2018. | | | | | | |
| UWAGI  Brak. | | | | | | |

Wskaźniki ilościowe charakteryzujące moduł/przedmiot:

|  |  |
| --- | --- |
| Szacunkowa sumaryczna liczba godzin pracy studenta (kontaktowych i pracy własnej) niezbędna dla osiągnięcia zakładanych dla zajęć efektów uczenia się - na tej podstawie należy wypełnić pole ECTS: | **120 h** |
| Łączna liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich lub innych osób prowadzących zajęcia: | **2 ECTS** |

Tabela zgodności kierunkowych efektów uczenia się z efektami przedmiotu:

|  |  |  |  |
| --- | --- | --- | --- |
| kategoria efektu | Efekty uczenia się dla zajęć: | Odniesienie do efektów dla programu studiów dla kierunku | Oddziaływanie zajęć na efekt kierunkowy\*) |
| Wiedza 01 | zna budowę systemu rozproszonego, potrafi podać jego elementy oraz przykłady oraz klasyfikację | K\_W07 | 2 |
| Wiedza 02 | zna i rozumie specyficzne cechy systemu rozproszonego | K\_W10 | 2 |
| Wiedza 03 | potrafi opisać operacje wykonywane w systemie rozproszonym, rozumie ich znaczenie | K\_W10 | 2 |
| Wiedza 04 | umie programować w języku wykonywanym na pośredniej warstwie oprogramowania | K\_W07 | 2 |
| Wiedza 05 | zna i rozumie zasady komunikacji klient-serwer, | K\_W10 | 2 |
| Wiedza 06 | zna rodzaje oraz cechy serwerów wykorzystywanych w systemie rozproszonym | K\_W10 | 2 |
| Umiejętności 01 | umie zaimplementować różnorodne technologie komunikacyjne w systemie rozproszonym | K\_U17 | 3 |
| Umiejętności 02 | umie dokonać optymalizacji komunikacji między komponentami systemu rozprosoznego, | K\_U20 | 2 |
| Umiejętności 03 | potrafi opracować system rozproszony z wykorzystaniem przeglądarki internetowej, po stronie klienta | K\_U17, K\_U18 | 3 |
| Kompetencje 01 | potrafi zdiagnozować wady systemu rozproszonego | K\_K03 | 3 |
| Kompetencje - |  |  |  |

\*)

3 – zaawansowany i szczegółowy,

2 – znaczący,

1 – podstawowy,